# Smart Email intake using AI models to analyze and route

02.01.2023

Recently we have had five different customer engaged at our Microsoft Technology Center targeting similar challenges with a common pattern to enable smart Email intake to analyze, route and response. During a sequence workshops staring with envisioning to identify and validate the business value, a architecture design workshop to design the solution and establish a sandbox environment for the follow-up Hackathon to implement the solution within the customer environment.

## Business scenario

Capture 500K plus Emails per year to intake, analyze and route them to operation teams. High level assessment shows the bandwidth of their challenges.

|  |  |  |
| --- | --- | --- |
|  | Simple | Complex |
| Email per year | 100’000 | 500’000 |
| Agents managing responses | 30 | 150 |
| Business analyst managing and improving the business process | 2 | 10 |
| First response rate | 50% | 20% |
| Average resolution time | 15min | 1h |
| Language supported | 85% one main language, rest various languages | 60% German, 30% French, 10% one or more additional languages |

## Customer challenges

Based on the customer envisioning we can summarize their challenges:

* We’re getting a lot unstructured Emails from our customers and partners triggering a business process to manage existing contracts or services.
* We want to analyze Email automatically and categorize the Email based on the product and customer intent.
* We want to enrich the Email with business context based on the target product and customer intent to increase our agent effectiveness and efficiency.
* We want to assign the analyzed Email to the best skilled and available agent.
* A majority of those Emails are generating replay or forward chit chats between sender, receivers and fulfiller.
* We need to support multi-languages, as customer or partner Emails are in English, French, Italian or German.
* We want to reduce the TCO of the target solution and enable business analyst to extend and drive on going changes.

## Table Description automatically generatedArt-of-the possible discovery

Figure 1 High-level functional process overview with domain options

As most customers already have an existing Microsoft 365 tenant, we can focus the art of possible discovery to the included Power Platform cloud services and cross-check if there other capabilities to reduce implementation time and TCO. For example:

* Including existing customer application and AI models via API management.
* Use of Power Platform and/or Dynamics 365 Premium services where it makes sense based on business value.

During the discussion we break down Smart Email intake process into six functional domains with different implementation and extension options. With this we can chain the functional domains based on customers objectives and requirement.

## Make or buy decision

My principle is use **make** if we have a competitive advantage, otherwise **buy**. For example the smart part (AI based) of the solution is make (bring you own model, train model with your data), rest make or buy. With that in mind we used a mix of Dynamics 365 Customer Services and Power Platform functionality. To reduce cost to buy and maintain, and accelerate implementation.

|  |  |  |
| --- | --- | --- |
| Functionality | Power Platform | Dynamics 365 |
| Email Intake | Power Automate flow using Office Outlook 365 Connector requires manually to handle Email details and attachment to create Email Activity assigned to Case. No automatic Tracking of response Email. | Automatic record creation rule supports Case, Contact and Email Activity creation including attachments and tracking of response Email |
| Routing to Queue | Power Automate flow maintaining the routing decision tree and case handling. | Unified Routing and Work Distribution supports rules and work distribution via Business Analyst without coding using configuration. |
| Forward Email using Email Tracking | No support of Email Tracking, when forwarding Email to other users/teams mailbox. | Dynamics 365 Outlook app supports Email tracking and context within Outlook.  Dynamics 365 automatically record creation from Email supports tracking as well. |
| Case Management | Power Apps requires custom build of Model Driven app and Case to maintain the process | Dynamics 365 Customer Services includes Case Management closed loop management out-of-the-box |
| Security Roles | Power Apps requires custom build of Security Roles to ensure access permission. | Dynamics 365 Customer Service includes predefined security roles for Agent and Service Owners. |

A quick business value justification for an assumed scenario with 500'000 Mails per year, 160 Agents managing responses using Dynamics 365 Team members, 10 Business Analysts managing process & models using Dynamics 365 Customer Services, with 3 Power AI Builder models used, turns out 20% cost benefit against Power Platform Premium licensing.

## Make it smart with AI models

Key component are the AI based models to analyze customer intent and context. Power AI Builder is a perfect starting point and we used the following prebuilt and custom models.

### Graphical user interface, application, Teams Description automatically generatedLanguage detection prebuilt model

Figure 2 Email language detection within the automatically record creation flow

The language detection prebuilt model identifies the predominant language of a text document. The model analyzes the text and returns the detected language and a confidence score from 0 through 1.

We used it to identify the language of the Email content to steer the after processing either translate it to a supported language or use language specific pre trained models for category classification or entity extraction.

AutomaticRecordCreationEmailToCase\_08

### Text translation prebuilt modelGraphical user interface, application, Teams Description automatically generated

The text translation prebuilt model translates your text data in real time across more than 60 languages. The text translation model can also detect the language of the text data you want to translate.

We used it to translate Emails into supported languages.

TranslateCasesToTargetLanguage\_11

### Sentiment analysis prebuilt model

The sentiment analysis prebuilt model detects positive or negative sentiment in text data. The scores and labels can be positive, negative, or neutral.

We used it for negative customer intent to escalate it.

### Graphical user interface, application Description automatically generatedCategory Classification custom model

Category classification is one of the fundamental natural language processing (NLP) challenges. With category classification, you can identify text entries with tags.

We used it to defined the following tags based on product and customer intent

* Pension fund; Withdrawal; Admission; Wage change;
* Vehicle Insurance; Deductible;

AIBuilderCreateCaseCategoryClassificationModel\_15

### Graphical user interface, application, Teams Description automatically generatedEntity Extraction custom model

AI Builder entity extraction models recognize specific data in text that you target based on your business needs. The model identifies key elements in the text and then classifies them into predefined categories. This can help you transform unstructured data into structured data that's machine-readable.

We used it to identify Vehicle Number Plate, Car insurance, AHV (Swiss Social Number).

AIBuilderCaseEntityExtractionModel\_07

**Vehicle Number Plate**

* Vehicle with registration plate {BE1234}
* Self-hasty for the vehicle {ZH32589}
* Vehicle {AI3214} an accident with {JU6539}

**Car Insurance**

* I have {caused damage} with my {vehicle} license plate
* My {vehicle} AI3214. Can you clarify how high my {deductible} is?
* Unfortunately, I had an {car} {accident }and have a question about it.

**AHV Number**

* AHV number {756.7289.6900.16}
* Pension fund for contact AHV numbers {756.6257.7019.21}
* Insurance for {756.1177.1531.19}

### Diagram Description automatically generatedUnified Routing engine

Figure 3 Dynamics 365 Customer Service Unified Routing engine overview

Dynamics 365 Unified routing is an intelligent, scalable, and enterprise-grade routing and assignment capability that can direct the incoming work item to the best-suited queue and agent by adhering to work item requirements and matching them with the agent’s capabilities.

Unified routing has two main stages: classification and assignment.

In the classification stage, rules and machine learning (ML) models can be used to add information on the work item, which can be used to find the best-suited agent.

We used it to route cases using Language and Category Classification model, Category Tags for products and customer intent to Agent skills like English, German, Car Insurance, Pension Funds, Withdraw, etc.

### References

* [Language detection prebuilt AI model - AI Builder | Microsoft Learn](https://learn.microsoft.com/en-us/ai-builder/prebuilt-language-detection)
* [Text translation prebuilt AI model - AI Builder | Microsoft Learn](https://learn.microsoft.com/en-us/ai-builder/prebuilt-text-translation)
* [Sentiment analysis prebuilt AI model - AI Builder | Microsoft Learn](https://learn.microsoft.com/en-us/ai-builder/prebuilt-sentiment-analysis)
* [Overview of category classification model - AI Builder | Microsoft Learn](https://learn.microsoft.com/en-us/ai-builder/text-classification-overview)
* [Entity extraction custom AI model overview - AI Builder | Microsoft Learn](https://learn.microsoft.com/en-us/ai-builder/entity-extraction-overview)
* [Overview of unified routing | Microsoft Learn](https://learn.microsoft.com/en-us/dynamics365/customer-service/overview-unified-routing)

## Make it real Hackathon

We build this during a 3 to 4 day Hackathon with a customer team with 3-6 (business stakeholder, business analyst and IT/developer) members and deployed the MVP to production.

If you want to discover more, check out the set by step guide providing a starting point to do it by your own.

### Steps by step guide

1. Create a solution “Case Management”
2. Create a Automatically create Case records from Email ruleset
3. Create instant flow “TranslateCasesToTargetLanguage”
4. Automatically create Case records from Email
5. AI Builder category classification custom model
6. Extend Automatically create Case flow with AI Builder models
7. Configure Unified Routing based on Category Tags to Agents

## Conclusion

Within 3 to 5 workshop days we were able to analyze the business value, design a target solution architecture and implement a MVP addressing customer key challenges. Overall it was a great work experience together to make it reality with a mixed customer stakeholder team. If you’ve similar challenges reach out to us for a workshop at the Microsoft Technology Center.

# Step by step guide

## Step 1 - Create a solution “Case Management”

Solutions are used to transport apps and components from one environment to another or to apply a set of customizations to existing apps. A solution can contain one or more apps as well as other components such as site maps, tables, processes, web resources, choices, flows, and more.

Solutions are the mechanism for implementing application lifecycle management (ALM) in Power Apps and other Power Platform products, such as Power Automate.

In our case we need to customize the Case table to support language, category tagging and related translation content. In the following steps we’ll add the following columns:

* Category Tags, for category classification tagging
* Category Tags Translated, for the translated into common language category classification tagging
* Description Translated, for the description translated into the common language
* Language, for the identified language code of the description content
* Language Score, for the confidence score of the detected language

In addition we’ll customize an existing form to show the “Case Analytics” columns and add a new view to review the “Case Analytics” data.

To create a solution:

1. Sign in to Power Apps (<https://make.powerapps.com/>) and select Solutions from the left navigation.
2. Select New solution and then complete the required columns for the solution.

|  |  |  |
| --- | --- | --- |
| Field | Value | Description |
| **Display Name** | Case Management | The name shown in the list of solutions. You can change this later. |
| **Name** | CaseManagement | The unique name of the solution. This is generated using the value you enter in the Display Name column. You can edit this before you save the solution, but after you save the solution, you can’t change it. |
| **Publisher** | <Your Publisher> | You can select the default publisher or create a new publisher. We recommend that you create a publisher for your organization to use consistently across your environments where you will use the solution. |
| **Version** | 0.0.0.1 | Enter a number for the version of your solution. This is only important if you export your solution. The version number will be included in the file name when you export the solution. |

1. Select Create.
2. Select “Add existing” – “Table”
3. Search for “Case” and select the table
4. Select “Next”
5. Select “Select objects” as we limit the customization and dependency to a minimum
6. In Columns search for “Description” and select the column
7. In Forms search for “Case for Interactive experience” and select the column
8. Select “Add”
9. Now we’ve to add additional columns to manage Category Tags, Language, Language Score and Translated text for Description and Category Tags as following.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Display Name | Category Tags | Category Tags Translated | Description Translated | Language | Language Score |
| Data type | Single line of text | Single line of text | Multiple lines of text | Single line of text | Number |
| Format | Plain text | Plain text |  | Plain text | Float |
| Schema name | categorytags | categorytagstranslated | descriptiontranslated | language | languagescore |
| Minimum value |  |  |  |  | 0 |
| Maximum character count | 4’000 | 4’000 | 1’048’576 | 2 | 1 |
| Decimal places |  |  |  |  | 5 |

1. Select “New column” and add the five new columns base on the table above to the Case table. See results

Graphical user interface, application

Description automatically generated

Figure 4 Case table with five new columns for case analytics

1. Select “Forms” in the objects navigation
2. Select “Case for Interactive experiences” and choose edit
3. In “Details” tab add an new section called “Case Analytics” and add the five new created columns and Description to the section. See result

Graphical user interface, application

Description automatically generated

1. Select “Publish”
2. Now we need to add a new view to review the Case Analytics. Select “Views” in the objects navigation
3. Select “New view” and name it “Active Case Analytics”

Graphical user interface, application, Teams

Description automatically generated

1. Select “Create”
2. Add the five columns and Description to the new view. Change Sort and Filter as needed. See result

Graphical user interface, text, application, email

Description automatically generated

1. Select “Publish”
2. Go back and select in the Objects navigation “All”. Select “Publish all customizations”
3. Now we can check if the customization is correct adapted. Open “Customer Service Hub”, select “Cases”.
4. Change view to “Active Case Analytics”

Graphical user interface, text, application

Description automatically generated

1. Select and open a case

Graphical user interface, application

Description automatically generated

Now we set the foundation to analysis cases. Next step is now to configure the Automatically create Case records from Email ruleset described in step 2.

### Reference

* [Solutions in Power Apps - Power Apps | Microsoft Learn](https://learn.microsoft.com/en-us/power-apps/maker/data-platform/solutions-overview)

## Step 2 - Automatically create Case records from Email

In the first step we laid the foundation, now let’s monitor mailboxes to intake Emails and prepare them for smart analysis.

You can automatically create or update system or custom records from incoming activities, such as emails, social activities, and custom activities. In this section, you'll learn about creating rules for automatically creating records for cases from incoming emails.

### Prerequisites

Make sure that the following prerequisites are met:

* Permissions, roles, and the Power Automate license to create automatic record creation rules.
* Information on the queues for which you want to create the rules. In our case we use [support@ursruegg.com](mailto:support@ursruegg.com)

### Configure rules for creating or updating records automatically

You can configure a rule that when active will be run for incoming emails. By using the feature to create rules, you can define the conditions for when a rule can be run.

You can configure the rules in the Customer Service admin center or Customer Service Hub app.

1. Open Customer Service Hub app and select Service Management.
2. Select Automatic record creation and update rules in Case Settings. The Record Creation and Update Rules page is displayed.
3. Select New. The New Record Creation and Update Rule page is displayed.
4. On the Basic tab, in Step one details, enter the following details:
   1. Rule name: Enter a name for the rule 🡪 “Support Email to Case”.
   2. Queue to monitor: Select a queue on which the rule will be activated. For email activity, you must select a queue to be able to activate the rule. 🡪 Support queue
   3. Activity type to monitor: Select Email in the list.

See result

Graphical user interface, application

Description automatically generated

1. Select Save. The Step two: conditions to evaluate and actions to take area is enabled.
2. In Step two: conditions to evaluate and actions to take, select New. The Condition builder page is displayed.

Graphical user interface, text, application, email, Teams

Description automatically generated

1. Perform the following steps:
   1. In Condition > Condition name, enter “Standard create case from email rule” name.
   2. In Condition that must pass to continue (pass if blank), select Add.
   3. Use the following options to define the criteria for the rule to be evaluated:
      1. Add row
      2. Add group
      3. Add related entity
2. In Actions to take > Record to create, make sure the value is “Case”. A case will be created if the conditions specified are met for the email activity.
3. In Configure in Power Automate, select Save and open in Power Automate. The Power Automate application opens in a new tab where you can configure criteria that must be evaluated for the email activity.
4. In the Power Automate tab we need to configure the steps to identify the language and add the results to the case record.
5. Add the following additional variables steps

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Initialize variable AI Builder Language | Initialize variable AI Builder Language Confidence Score | Initialize variable AI Builder Category Tag |
| Name | ai\_builder\_language | ai\_builder\_language\_confidence\_score | ai\_builder\_language\_category\_tag |
| Type | String | String | String |
| Value | en | 0 | none |

See results

1. Add a Flow Scope step and name it “Analyse Email using AI Builder models”
2. Add a Flow Step “Content Conversation” and name it “Convert email to plain text by inputting 'Description' dynamic content”
3. Add a Flow Scope and name it “AI Builder Language Detection”
4. Add a AI Builder Step “Detect the language being used in text” and name it “Detect the language being used in text”
5. Set Test to the result of the “Content Conversation” output
6. Add a Flow Step “Set variable” and name it “Set variable AI Builder Language”. Choose the “ai\_builder\_language” variable for Name and for Value “Language” result from the AI Builder.
7. Add a Condition step to check if detected language confidence score is valid <=1 and >=0.
8. Add in “If yes” a new Flow step Set variable and name it “Set variable AI Builder Language Confidence Score”. Set Name equal ai\_builder\_language\_confidence\_score and Value equal User Input Confidence score of the AI Builder output.
9. Go to step “Create a record (don't rename this step)” and open it to add additional fields to create the case record containing detected language and confidence score.
10. Set Description field to the output value of the “Content Conversation” output
11. Set Language field to the variable ai\_builder\_language
12. Set Language Score field to ai\_builder\_language\_confidence\_score
13. Save it and test it by sending a first Email sample. Check if a case is created with associated contact and email activity.

### Reference

* [Automatically create or update records in Dynamics 365 Customer Service | Microsoft Learn](https://learn.microsoft.com/en-us/dynamics365/customer-service/automatically-create-update-records?tabs=customerservicehub)
* [Get started with Power Automate (contains video) - Power Automate | Microsoft Learn](https://learn.microsoft.com/en-us/power-automate/getting-started)

## Step 3 - Create instant flow to translate sample data into target language

As we want to support three languages, English, French and German to analyze the Email content, we need to have sample data for each supported language. AI Builder has a Text translation prebuilt model and we can use it to translate the Case Description into a selected target language.

|  |  |
| --- | --- |
| Column | Description |
| Description | Holds the plan text email content. |
| Description Translated | Holds the translated to target language “Description” to be used in the AI Builder Category Classification model |
| Category Tags | Holds the analyzed category tags of the Description text.  In a first iteration you will add the Category Tags manually. After you’ve trained models, we’ll include the trained model into the “Automatically create Case records from Email” flow to analyze the Email content and get the Category Tags based on the trained model. |
| Category Tags Translated | Holds the Category Tags to be used in the AI Builder Category Classification model |

Consider how to deal with single or multi-language model support

* For a multi-language scenario I recommend to keep the Category Tags in one common language, as we will use the Category Tags later to route the Case using Routing Rules to the skilled Agent. In our case we used English as the common language for tagging the categories.
* For a single-language scenario, I recommend to translate everything into one common language like English. And just use one common language model to analyze and route the cases.

### Prerequisites

Make sure that the following prerequisites are met:

* Permissions, roles, and the Power Automate license to create automatic record creation rules.
* Case entity prepared as described in “Create a solution Case Management” before

### Create Translate Cases to Target Language instant flow

The helper flow should be triggered manually passing a target language code value to translate the Description into the target language and persist it into Description Translated for later used to train the AI Builder model for different language. The building blocks are:

1. Define the used variables
2. Check user input
3. Get list of cases
4. Translate content to target language

Graphical user interface, application, Teams

Description automatically generated

Now let’s create the helper flow

1. Sign in to Power Apps (https://make.powerapps.com/), and then select Flows > Cloud flows.
2. Select “New flow” and select “Instant cloud flow”
3. Add the following additional variables steps

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Initialize variable Target Language | Initialize variable Translated Description | Initialize variable Translated Category Tag | Initialize variable AI Builder Language | Initialize variable AI Builder Language Confidence Score | Initialize variable Case Description Translated |
| Name | target\_language | translated\_description | translated\_category\_tag | ai\_builder\_language | ai\_builder\_language\_confidence\_score | case\_description\_translated |
| Type | String | String | String | String | Float | String |
| Value | en | none | none | en | 0 | none |

See results

Graphical user interface, application, Teams

Description automatically generated

1. Add a Flow Scope step and name it “Check user input”
2. Add a Condition step to check if user input is a valid supported target language code.
3. Add in “If yes” a new Flow step Set variable and name it “Set variable Target Language to User Input”. Set Name equal target\_language and Value equal User Input target\_language.

Graphical user interface, application

Description automatically generated

1. Add a Flow Scope step and name it “Get list of cases”
2. Add a Dataverse List rows action and name it “List cases”
3. Select Table name “Cases”
4. Add Fetch Xml Query to get only active cases with category tags. See below sample

<fetch version="1.0" output-format="xml-platform" mapping="logical" distinct="false">

<entity name="incident">

<attribute name="title" />

<attribute name="incidentid" />

<attribute name="ticketnumber" />

<attribute name="casetypecode" />

<attribute name="ur\_categorytags" />

<attribute name="customerid" />

<attribute name="description" />

<attribute name="ur\_language" />

<attribute name="ur\_languagescore" />

<attribute name="adx\_resolution" />

<attribute name="subjectid" />

<attribute name="msdyn\_incidenttype" />

<attribute name="statecode" />

<attribute name="ur\_descriptiontranslated" />

<attribute name="ur\_categorytagstranslated" />

<order attribute="title" descending="false" />

<filter type="and">

<filter type="and">

<condition attribute="statecode" operator="in">

<value>0</value>

<value>1</value>

<value>2</value>

</condition>

<condition attribute="description" operator="not-null" />

<condition attribute="ur\_categorytags" operator="not-null" />

</filter>

</filter>

</entity>

</fetch>

1. Add a Flow Scope step and name it “Translate content to target language”
2. Add a AI Builder Step “Detect the language being used in text” and name it “Detect the language being used in case description”
3. Add a Flow Step “Set variable” and name it “Set variable AI Builder Language”. Choose the “ai\_builder\_language” variable for Name and for Value “Language” result from the AI Builder.

Graphical user interface, application

Description automatically generated

1. Add a Condition step to evaluate the detected language confidence score is a valid result between 0 and 1.
2. Add in “If yes” a new Flow step Set variable and name it “Set variable AI Builder Language Confidence Score”
3. For Name choose the variable “ai\_builder\_language\_confidence\_score” and for Value add the AI Builder Confidence Score result.
4. Add in “If no” a new Flow step Set variable and name it “Set variable AI Builder Language Confidence Score”
5. For Name choose the variable “ai\_builder\_language\_confidence\_score” and for Value add 0.

Graphical user interface, application, Teams

Description automatically generated

1. Add a Condition step to evaluate the detected language is a valid result en, fr, de

Graphical user interface, application

Description automatically generated

1. Add in “If yes” a new Flow step switch to evaluate the supported target language model. Set On equal to target\_language variable
2. Add a Switch cases for English language and name it “Language equal english”
3. Set Equals to en
4. Add a AI Builder Step “Translate text into another language” and name it “Translate description to english”.
5. Set Text to Case Description and Translate to English
6. Add a Flow Step “Set variable” and name it “Set variable Translated Description for english”. Choose the “translated\_description” variable for Name and for Value “Translated text” result from the AI Builder.
7. Add a AI Builder Step “Translate text into another language” and name it “Translate category tag to english”.
8. Set Text to Case Category Tags and Translate to English
9. Add a Flow Step “Set variable” and name it “Set variable Translated Category Tag for english”. Choose the “translated\_category\_tag” variable for Name and for Value “Translated text” result from the AI Builder. See result
10. Graphical user interface, application, Teams

    Description automatically generated
11. To support other languages too, repeat the steps 21 to 29 for French and German.
12. Close the switch action
13. Add a Dataverse Step “Update a row” and name it “Update current case”
14. Set Table name equal “Cases”, Row ID equal “Case”, Category Tags Translated to translated\_category\_tag variable and Description Translated to translated\_description.
15. Graphical user interface, application, Teams

    Description automatically generated
16. Select “Save” to persist all changes you made to the flow.
17. Select “Test” to test your flow with User Input with a supported language code.

Now we’ve helper flow to simple translate the Case description into any target language. We will use this helper flow to prepare the sample data for the AI Builder models for category classifications.

## Step 4 - AI Builder category classification custom model

The volume of text data is increasing exponentially for organizations. Channels such as email, documents, and social media contribute increasing amounts of text data. This data carries valuable information that—when extracted and acted on—helps you provide better services to your customers.

Now we can build custom AI models to analyze customer intent and context using Power AI Builder category classification custom models. We want to support multi languages and there are two approaches to consider:

1. Use one common language model – In this case you need to translate all emails into the common language, for example English and use the translated content for the classification models.
2. Use for each language dedicated models – In this case you need to establish for each supported language a dedicated and trained model.

Booth approaches has their advantages and disadvantages

|  |  |  |
| --- | --- | --- |
| Comparison | Common Model | Dedicated Models |
| Advantages | * Supports any languages * Simplifies the maintenance to train and retrain the model | * Higher accuracy to identify customer intent. |
| Disadvantages | * Accuracy to identify customer intent. Some terms/sentence are vey specific to the language | * Higher maintenance to train and retrain the models |

### Prerequisites

* This model requires the training data to be available within a Dataverse table. In our case we’ll use Cases and have loaded some test data.
* Make sure your administrator has assigned you a security role with Read privilege for the table that has the training data.

### Data preparation

The training data used to train the model from the Dataverse table should conform to the following:

* Store text and tags as two columns in the same table. Each row must have data in the Text column.
* You can provide one or more tags to data in the same row in the Text column.
* If you've identified multiple tags within the text sample, provide them as delimited text in the Tags fields. We use semicolons (;) as tag separators. For example:

|  |  |
| --- | --- |
| Description | Category Tags |
| Please make the following resignation.  Contract  0DB78F  AHV numbers  756.9950.7311.96  Hansrudolf Muster  Obere Musterstrasse 18b  8000 Zurich | Pension fund; Withdrawal; |
| Hello  Unfortunately, I had an accident and have a question about it.  My vehicle AI3214. Can you clarify how high my deductible is?  Many thanks for the prompt clarification  Best regards,  Marc Brühlhard  Bahnofstrasse 13  8001 Zurich | Vehicle Insurance; Deductible; |

* Make sure to have a minimum of 10 distinct text samples for each tag to be extracted. Tags with fewer than 10 samples won't be trained.
* If Cases has been tagged in fewer than 10 rows in the data, it will be ignored. The model won't be trained to categorize data for that tag.
* For every tag that is used, provide a minimum of 10 text samples where it isn't used.

### Steps to prepare sample data

1. In our case we can send Email with sample text to [support@ursruegg.com](mailto:support@ursruegg.com) and the automatically create records rule creates for each Email a new case.
2. Prepare and send at least 10 Emails with sample data
3. Open “Customer Service Hub” and select cases.
4. Open the view “Active Case Analytics” and check if the cases with the sample data is there.
5. Select “Export to Excel” – “Open in Excel Online”
6. Update the Category Tag for each case. In our case we defined the following tags based products and customer intent
   * Pension fund; Withdrawal; Admission; Wage change;
   * Vehicle Insurance; Deductible;

Graphical user interface, application, table, Excel

Description automatically generated

1. Select “Save” and the updated cases including the category tags will be imported
2. Go back to the view “Active Case Analytics” and check if the category tags are updated

**Keep in mind**

* A table must have at least two tags, and each one must have 10 text samples.
* You can define up to 200 distinct tags. Each tag is a category that will be identified and extracted from the given text.
* Each sample of text data must have fewer than 5’000 characters.
* In our case we use the Case Description column with configured max size of 2’000 characters out-of-the-box.
* If you need to support more than 2’000 characters, you can create a new column of “Multiple line of text” – “Plain text” that support max size of 1’048’576 characters. Review the “Description Translated” column.

### Create a category classification custom model

Now that you have your training data in Microsoft Dataverse, you can create a new model and configure it.

1. Sign in to Power Apps (https://make.powerapps.com/), and then select AI Builder > Explore.
2. Select Text.

Graphical user interface, application

Description automatically generated

1. Select Category classification - Classify texts into custom categories.
2. Read the Classify texts into custom categories page, and then select Get started.
3. The Category classification tile step by step window opens.

Graphical user interface, application

Description automatically generated

1. Change the model name to “Case Category Classification English”
2. Choose “Select text”, select the table “Case”, select the column “Description Translated” where your training text is stored, and then choose Select column.

Graphical user interface, application, Teams

Description automatically generated

1. Select Next. See result

Graphical user interface, text, application, email, Teams

Description automatically generated

1. Choose Select tags, select the column “Category Tags Translated” where the tags are stored, and then choose Select column.

Graphical user interface, application, Teams

Description automatically generated

1. (If not pre-selected) Select the separator you used for your tags, in our case “Semicolon” and then select Next.

Graphical user interface, application

Description automatically generated

1. Review your text and tags to verify the data and the configuration you applied, and then select Next.

Graphical user interface, text, application

Description automatically generated

1. Select the language “English” you want to use for training, and then select Next.

Graphical user interface, application

Description automatically generated

1. To begin training your category classification model, review your configuration, and then select Train. Graphical user interface, application, Teams

   Description automatically generated
2. When it's ready, you'll be notified.

Graphical user interface, application

Description automatically generated

1. After each training, AI Builder uses the test dataset to evaluate the quality and accuracy of your AI model. A summary page for your model shows your model training results, including a Performance score.

Graphical user interface, application, Teams

Description automatically generated

1. You can also select Quick Test to assess the quality of the model. Just enter text that you want to tag.

Graphical user interface, application

Description automatically generated

1. As a result you should get category tags with a confidence score

That's it! Now you have the first trained AI model. Now we need to establish for each supported language a dedicated Category Classification model. In our case we want to support English, French and French.

### Reference

* [Overview of category classification model - AI Builder | Microsoft Learn](https://learn.microsoft.com/en-us/ai-builder/text-classification-overview)
* [Language detection prebuilt AI model - AI Builder | Microsoft Learn](https://learn.microsoft.com/en-us/ai-builder/prebuilt-language-detection)
* [Text translation prebuilt AI model - AI Builder | Microsoft Learn](https://learn.microsoft.com/en-us/ai-builder/prebuilt-text-translation)

## Step 5 - Extend Automatically create Case flow with AI Builder models

Now we can extend the standard out-of-the-box flow to analyze the content. We need to detect the Language of the Email content.

1. Open Customer Service Hub app and select Service Management.
2. Select Automatic record creation and update rules in Case Settings. The Record Creation and Update Rules page is displayed.
3. Select “Support Email to Case” and open it.
4. On the Basic tab, in Step two select “In Configure in Power Automate”, select open in Power Automate.
5. Add the following additional variables steps

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Initialize variable AI Builder Language | Initialize variable AI Builder Language Confidence Score | Initialize variable AI Builder Category Tag |
| Name | ai\_builder\_language | ai\_builder\_language\_confidence\_score | ai\_builder\_language\_category\_tag |
| Type | String | String | String |
| Value | en | 0 |  |

See results

Graphical user interface, application, Teams

Description automatically generated

1. Add a Flow Scope step and name it “Analyze Email using AI Builder models”
2. Add a “Html to text” to convert the Email HTML content to plain text and name it “Convert email to plain text by inputting 'Description' dynamic content”
3. Add “Description” column to the content field

Graphical user interface, application

Description automatically generated

1. Add a Flow Scope step and name it “AI Builder Language Detection”
2. Add a AI Builder Step “Detect the language being used in text” and name it “Detect the language being used in text”
3. Add the result of the Html to text conversion to the Text of the AI Builder step
4. Add a Flow Step “Set variable” and name it “Set variable AI Builder Language”. Choose the “ai\_builder\_language” variable for Name and for Value “Language” result from the AI Builder.
5. Add a Condition step to evaluate the detected language confidence score is a valid result between 0 and 1.
6. Add in “If yes” a new Flow step Set variable and name it “Set variable AI Builder Language Confidence Score”
7. For Name choose the variable “ai\_builder\_language\_confidence\_score” and for Value add the AI Builder Confidence Score result. See result

Graphical user interface, application, Teams

Description automatically generated

1. Add a Flow Scope step and name it “AI Builder Case Category Classification Detection”. Leave it empty as a placeholder. We will come back to extend this part after we’ve created and trained the AI Builder category classification models.
2. Select the Dataverse Step “Create a record (don't rename this step)” and update the Description equal to result of the Html to text conversion, Language equal to ai\_builder\_language variable and Language Score equal to ai\_builder\_language\_confidence\_score. See result

Graphical user interface, text, application, email

Description automatically generated

1. Select “Save” to update the flow with all your changes.
2. Go back to the “Record Creation and Update Rule Item”
3. Select Save & Close. The condition builder is closed.
4. In Step three: additional actions to take after matching with a condition, select an option based on your requirement for Automatically reply to email.
5. If you select Yes, select an email template in the Select email template list box.

Now we’ve created the rule set and are ready to test it. Send some example mail to the Email address and analyze the result in “Customer Service Hub” – “Cases” – “Active Case Analytics”.

## Step 6 - Configure Unified Routing based on Category Tags to Agents

Add skills for each used Category Tag and assign skills to Agents (Users) experience.

In our case we defined the following category tags based products and customer intent

o Pension fund; Withdrawal; Admission; Wage change;

o Vehicle Insurance; Deductible;

Fahrzeugversicherung; Selbsbehalt; Unfallversicherung; Haftpflicht; Pensionskasse; Eintritt; Austritt; Anfrage; Adressänderung; Lohnänderung

## AI Builder category classification custom model

AI Builder entity extraction models recognize specific data in text that you target based on your business needs. The model identifies key elements in the text and then classifies them into predefined categories. This can help you transform unstructured data into structured data that's machine-readable. You can then apply processing to retrieve information, extract facts, and answer questions.

### Requirements and limitations

Keep in mind the current supported data format and languages.

* Data format: Documents can't exceed 5,000 characters.
* Supported languages: English, Chinese-Simplified, French, German, Portuguese, Italian, Spanish

### Prerequisites

* This model requires the training data to be available within a Dataverse table. In our case we’ll use Cases and have loaded some test data.
* Make sure your administrator has assigned you a security role with Read privilege for the table that has the training data.

### Create an entity extraction custom model

To create your custom entity extraction model:

* Provide at least 10 examples of your text data.
* Review the results from existing, prebuilt entities.
* Refine your results by creating your own custom tables or modifying existing, prebuilt tables.
* Review your model, and train it.
* Evaluate your model (optional).
* Upload examples of your text data

You need at least 10 data examples to start customizing the model. AI Builder will identify some tables out of the box. You can customize your model by creating new entity types with a small set of examples or by modifying the existing entity types.

### Create Extract custom entities from cases

We are building the model on top of the same sample cases we prepared earlier in Microsoft Dataverse, you can create a new model and configure it.

1. Sign in to Power Apps (https://make.powerapps.com/), and then select AI Builder > Explore.
2. Select Text.  
   Graphical user interface, application

   Description automatically generated
3. In the left pane, select AI Builder > Explore, and then select Extract custom entities from your text

Graphical user interface, application, Teams

Description automatically generated

1. Select “Get started” and the step by step dialog shows up
2. Choose the Table = Case and Column = Description Translated

Graphical user interface, text, application, email

Description automatically generated

1. Choose the Language based on the prepared data. In our case French

Graphical user interface, application, Teams

Description automatically generated

1. Now the case description are analyzed using the out-of-the-box entities

Graphical user interface, application

Description automatically generated

1. Review the automatically identified entities based on your case data
2. As we want to identify the product and customer intent we will create new custom entities for

|  |  |
| --- | --- |
| Entity | Descriptions |
| VehicleNumberPlate | Number plate of the car |
| CarInsurance | Car insurance product identification |
| AHVNumber | Swiss social number |
| PolicyIntention | Policy intention identification |

1. Select “New entity” and name it “VehicleNumberPlate”
2. Select “Add new example” and add at least five example phrase with the term in {} as following

Enter a name for your model, and then select Create.

Select the Microsoft Dataverse Table and the Column that contains your data.

Here's an example of a selected table and column:

Add data.

Review and refine entities

AI Builder will automatically extract the prebuilt entity types from your uploaded examples. If the results are satisfactory, you don't need to customize a model and can use the prebuilt entity extraction model out of the box.

You can customize your entity extraction model in these ways:

Create a new entity type.

You need to provide at least five examples to create a new entity type. For example, to create a new entity type named size, you can add an example like "The suitcase was {large}." The braces designate that "large" is of entity type size.

Modify an existing entity type.

You need to add least five examples to an existing entity type to modify it.

Deselect any prebuilt entities that you don't want to include in the model.

You can select/deselect prebuilt entity types from the settings.

#### Case Entity Extraction Model

**VehicleNumberPlate**

Fahrzeuges mit Kontrollschild {BE1234}

Selbstbehalt für das Fahrzeug {ZH32589}

Fahrzeug {AI3214}

einen Umfall mit {JU6539}

Selbstbehalt für {FR2154}

Fahrzeugversicherung {SH86445}

Fahrzeug mit Kontrollschild {SH86445}

Selbstbehalt für Kontrollschild {FR2154}

Vehicle with registration plate {BE1234}

Self-hasty for the vehicle {ZH32589}

Vehicle {AI3214} an accident with {JU6539}

Deductible for {FR2154}

Vehicle Insurance {SH86445}

Vehicle with registration plate {SH86445}

Deductible for control plate {FR2154}

Véhicule avec plaque d’immatriculation {BE1234}

Auto-précipitation pour le véhicule {ZH32589}

Véhicule {AI3214} un accident avec {JU6539}

Franchise pour {FR2154}

Assurance véhicule {SH86445}

Véhicule avec plaque d’immatriculation {SH86445}

Franchise pour plaque de contrôle {FR2154}

**CarInsurance**

What is my {deductible} for {vehicle} with registration plate

I have {caused damage} with my {vehicle} license plate

My {vehicle} AI3214. Can you clarify how high my {deductible} is?

Unfortunately, I had an {car} {accident }and have a question about it.

What is my {deductible} for {vehicle policy}

I have a question regarding my {vehicle} {insurance}

I had an {accident} with JU6539 and would like to know my car {policy} coverage

What is my {deductible} for {vehicle} with registration plate

I have {caused damage} with my {vehicle} license plate

Qu’est-ce que mon {deductible} pour {vehicle} avec plaque d’immatriculation

J’ai {causé des dommages} avec ma plaque d’immatriculation {véhicule}

Mon {véhicule} AI3214. Pouvez-vous préciser le montant de ma {franchise}?

Malheureusement, j’ai eu une {voiture} {accident} et j’ai une question à ce sujet.

Quelle est ma {franchise} pour {police véhicule}

J’ai une question concernant mon {véhicule} {assurance}

J’ai eu un {accident} avec JU6539 et j’aimerais connaître ma couverture de voiture {police}

Quelle est ma {franchise} pour {véhicule} avec plaque d’immatriculation

J’ai {causé des dommages} avec ma plaque d’immatriculation {véhicule}

**AHVNumber**

AHV numbers {756.7289.6900.16}

My AHV number is {756.9950.7311.96}

Pension fund for contact AHV numbers {756.6257.7019.21}

AHV number {756.1177.1531.19}

Insurance for {756.1177.1531.19}

My AHV number is {756.6317.9277.73}

Pension fund admission for {756.6317.9277.73}

AHV Number {756.2863.5749.59}

Insurance for {756.1558.4737.99}

AHV Nummer {756.7289.6900.16}

Meine AHV nummer ist {756.9950.7311.96}

Pensionskasse für Vertrag mit AHV nummers {756.6257.7019.21}

AHV nummer {756.1177.1531.19}

Versicherung für {756.1177.1531.19}

Die AHV ist {756.6317.9277.73}

BVG Neuzugang für {756.6317.9277.73}

AHV Nummer {756.2863.5749.59}

Versicherung für {756.1558.4737.99}

Numéros AVS {756.7289.6900.16}

Mon numéro AVS est {756.9950.7311.96}

Caisse de pension pour les numéros de contact AVS {756.6257.7019.21}

Numéro AVS {756.1177.1531.19}

Assurance pour {756.1177.1531.19}

Mon numéro AVS est {756.6317.9277.73}

Admission à la caisse de retraite pour {756.6317.9277.73}

Numéro AVS {756.2863.5749.59}

Assurance pour {756.1558.4737.99}

**PolicyIntention**

{Pension fund} {admission} for 756.6317.9277.73

{Pension fund} {withdrawal} for 756.1558.4737.99

{Pension fund} {wage change} to CHF 120’000 for 756.1558.4737.99